UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/826,723	04/16/2004	Andreas S. Krebs	11884/502501	1351

53000   7590   06/02/2008
KENYON & KENYON LLP
1500 K STREET N.W.
WASHINGTON, DC 20005

EXAMINER
ABDUL-ALI, OMAR R

ART UNIT	PAPER NUMBER
2178	

MAIL DATE	DELIVERY MODE
06/02/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 March 2008*.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,5-10,12-17,19,21-26,28-34 and 36-39* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1, 2, 5-10, 12-17, 19, 21-26, 28-34, and 36-39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application

6)☐ Other: _____ .

DETAILED ACTION

The following action is in response to the Request for Continued Examination (RCE)

filed March 13, 2008. Amended Claims 1, 2, 5-10, 12-17, 19, 21-26, 28-34, and 36-39

are pending and have been considered below.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1, 2, 5, 6, 8-10, 12-14, 16, 17, 19, 21, 22, 24-26, 28-30, 32-34 and 36-39

remain rejected under 35 U.S.C. 103(a) as being unpatentable over Oni (US

2004/0133546) in view of Emmerichs (US 2003/0061482).

Claims 1, 9, and 33: Oni discloses a method of managing visibility of GUI components

in an application, comprising:

a. initializing the application (page 3, paragraph 43);

b. a user interface for displaying a user interface screen of the application

according to an applied one of the plurality of profiles (page 4, paragraph 62);

Oni discloses invoking a visibility manager for a plurality of profiles and displaying

a user interface of the application wherein display of the GUI components are

determined by the visibility manager (page 4, paragraph 62), but does not explicitly

disclose providing a user interface of the visibility manager for selection of one of a

plurality of visibility states for each of at least a subset of the GUI components.

Emmerichs discloses a similar method of managing visibility of GUI components in an

application that further discloses selecting options in a security manager (visibility

manager) that control the visibility states of GUI objects accessed by a user through a

user interface. Therefore it would have been obvious to one having ordinary skill in the

art at the time the invention was made to provide a user interface of the visibility

manager in Oni, because providing a selection interface was recognized as part of the

ordinary capabilities of one skilled in the art. One would have been motivated to provide

a user interface of the visibility manager in order to provide access to GUI components

based on access privileges.

Emmerichs further discloses building the user interface screen with all of the

components set as visible, then invoking the visibility manager to determine, based on

the selected visibility states of the applied profile, which of the GUI components of the

built user interface screen are to be set as not visible and revise the built user interface

screen based on the determination and displaying the revised user interface screen of

the application (page 6, paragraph 54). Specifically Emmerichs discloses displaying all

of the user interface components without first determining whether the user has access

to each widget. After the user attempts to access the widget, the software checks the

availability of the widget and notifies a user with an error message if access to the

widget is denied. Therefore, it would have been obvious to one having ordinary skill in

the art to include these limitations in Oni. One would have been motivated to revise the

user interface based on the determination of the visibility state of the applied profile in

order to provide an initial GUI to the user while the profile is loading.

Claim 2: Oni and Emmerichs disclose a method of managing visibility of GUI

components in an application as in Claim 1 above, and Oni further discloses:

      a. reading the plurality of profiles (page 4, paragraphs 63 and 67);

      b. processing the plurality of profiles (page 4, paragraphs 63 and 67);

      c. reading and processing a user configuration based on the plurality of profiles

(page 4, paragraphs 62 and 67);

      d. activating the applied profile (page 4, paragraph 65).

Claim 5: Oni and Emmerichs disclose a method of managing visibility of GUI

components in an application as in Claim 2 above, and Oni further discloses:

      a. selecting an identification of a particular GUI component (page 4, paragraph

67);

      b. locating the identification in a mapping table [dynamic repository] (page 4,

paragraph 67);

      c. checking a state of the particular GUI component (page 4, paragraph 69);

      d. comparing the state to the applied profile (page 4, paragraph 69);

      e. changing the state if not in agreement with the applied profile (page 4,

paragraph 69);

f. repeating locating the identification, checking the state, comparing the state,

and changing the state for any remaining identifications of additional GUI components

(page 4, paragraph 69).

Claim 6: Oni and Emmerichs disclose a method of managing visibility of GUI

components in an application as in Claim 5 above, and Oni further discloses:

a. the state is visible or not visible (page 4, paragraph 69).

Claim 8: Oni and Emmerichs disclose a method of managing visibility of GUI

components in an application as in Claim 1 above, and Oni further discloses:

a. reading the plurality of profiles (page 4, paragraph 63);

b. processing the plurality of profiles (page 4, paragraph 63);

c. reading and processing a user configuration based on the plurality of profiles

(page 4, paragraph 62);

d. activating the applied profile by:

1) selecting an identification of a particular GUI component (page 4,

paragraph 67);

2) locating the identification in a mapping table (page 4, paragraph 67);

3) checking a state of the particular GUI component (page 4, paragraph

69);

4) changing the state if not in agreement with the applied profile (page 4,

paragraph 69);

5) repeating locating the identification, checking the state, comparing the state, and changing the state for any remaining identifications of additional GUI components (page 4, paragraph 69).

Claim 10: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 9 above, and Oni further discloses:

a. read the one or more profiles upon initialization of the application (page 4, paragraph 63);

b. process the one or more profiles (page 4, paragraph 63);

c. read and process a user configuration based on the one or more profiles (page 4, paragraph 62);

d. activate a particular profile of the one or more profiles (page 4, paragraph 65).

Claim 12: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 10 above, and Oni further discloses:

a. select an identification of a particular GUI component (page 4, paragraph 67);

b. locate the identification in a mapping table (page 4, paragraph 67);

c. check a state of the particular GUI component (page 4, paragraph 69);

d. compare the state to the particular profile of the one or more profiles (page 4, paragraph 69);

e. change the state if not in agreement with the particular profile of the one or more profiles (page 4, paragraph 69);

Claim 13: <u>Oni</u> and <u>Emmerichs</u> disclose a system of managing visibility of GUI

components in an application as in Claim 12 above, and <u>Oni</u> further discloses:

      a. activating the particular profile of the one or more profiles further comprises

repeating locating the identification, checking the state, comparing the state, and

changing the state for any remaining identifications of additional GUI components (page

4, paragraph 69).

Claim 14: <u>Oni</u> and <u>Emmerichs</u> disclose a system of managing visibility of GUI

components in an application as in Claim 12 above, and <u>Oni</u> further discloses:

      a. the state is visible or not visible (page 4, paragraph 69).

Claim 16: <u>Oni</u> and <u>Emmerichs</u> disclose a system of managing visibility of GUI

components in an application as in Claim 9 above, and <u>Oni</u> further discloses:

      a. read the one or more profiles upon initialization of the application (page 4,

paragraph 63);

      b. process the one or more profiles (page 4, paragraph 63);

      c. read and process a user configuration based on the one or more profiles (page

4, paragraph 62);

      d. select an identification of a particular GUI component (page 4, paragraph 67);

      e. locate the identification in a mapping table (page 4, paragraph 67);

      f. check a state of the particular GUI component (page 4, paragraph 69);

g. compare the state to the particular profile of the one or more profiles (page 4, paragraph 69);

h. change the state if not in agreement with the particular profile of the one or more profiles (page 4, paragraph 69);

Claim 17: Oni discloses a system of managing visibility of GUI components in an application, comprising:

a. a processor;

b. use a visibility manager data structure in managing visibility of GUI components in a user interface of an application, the visibility manager data structure comprising a mapping table, one or more profiles and a user configuration identifying which of the one or more profiles is to be applied (page 4, paragraphs 65 and 67). Specifically, Oni discloses linking user interface elements with the repository of user profiles, and matching the profile of a new user with the profile of previously stored profiles for which combinations of user interface elements exist in the solution repository.

c. display the user interface screen of the application according to an applied one of the profiles using the application and the visibility manager (page 4, paragraph 65).

Oni discloses invoking a visibility manager for a plurality of profiles and displaying a user interface of the application wherein display of the GUI components are determined by the visibility manager (page 4, paragraph 62), but does not explicitly disclose providing a user interface of the visibility manager for selection of one of a

plurality of visibility states for each of at least a subset of the GUI components.

Emmerichs discloses a similar method of managing visibility of GUI components in an

application that further discloses selecting options in a security manager (visibility

manager) that control the visibility states of GUI objects accessed by a user through a

user interface. Therefore it would have been obvious to one having ordinary skill in the

art at the time the invention was made to provide a user interface of the visibility

manager in Oni, because providing a selection interface was recognized as part of the

ordinary capabilities of one skilled in the art. One would have been motivated to provide

a user interface of the visibility manager in order to provide access to GUI components

based on access privileges.

Emmerichs further discloses the application building the user interface screen of

the application with all of the GUI components set as visible and subsequently calling

the visibility manager, the visibility manager, responsive the calling: determining based

on the selected visibility states of the applied profile which of the GUI components of the

built user interface screen are to be set as not visible and revising the built user

interface screen based on the determination and the application displaying the revised

user interface screen of the application. (page 6, paragraph 54). Specifically

Emmerichs discloses displaying all of the user interface components without first

determining whether the user has access to each widget. After the user attempts to

access the widget, the software checks the availability of the widget and notifies a user

with an error message if access to the widget is denied. Therefore, it would have been

obvious to one having ordinary skill in the art to include these limitations in Oni. One

would have been motivated to revise the user interface based on the determination of

the visibility state of the applied profile in order to provide an initial GUI to the user while

the profile is loading.

Claim 19: Oni and Emmerichs disclose a system of managing visibility of GUI

components in an application as in Claim 17 above, and Oni further discloses:

a. read the one or more profiles upon initialization of the application (page 4,

paragraph 63);

b. process the one or more profiles (page 4, paragraph 63);

c. read and process the user configuration based on the one or more profiles

(page 4, paragraph 62);

d. activate the identified profile of the one or more profiles based upon the

mapping table (page 4, paragraph 67);

Claim 21: Oni and Emmerichs disclose a system of managing visibility of GUI

components in an application as in Claim 19 above, and Oni further discloses:

a. select an identification of a particular GUI component (page 4, paragraph 67);

b. locate the identification in a mapping table (page 4, paragraph 67);

c. check a state of a specific GUI component in the application (page 4,

paragraph 69);

d. compare the state to the particular profile of the one or more profiles (page 4,

paragraph 69)

e. change the state if not in agreement with the identified profile of the one or

more profiles (page 4, paragraph 69);

Claim 22: <u>Oni</u> and <u>Emmerichs</u> disclose a system of managing visibility of GUI

components in an application as in Claim 21 above, and <u>Oni</u> further discloses:

a. the state is visible or not visible (page 4, paragraph 69).

Claim 24: <u>Oni</u> and <u>Emmerichs</u> disclose a system of managing visibility of GUI

components in an application as in Claim 19 above, and <u>Oni</u> further discloses:

a. read the one or more profiles upon initialization of the application (page 4,

paragraph 63);

b. process the one or more profiles (page 4, paragraph 63);

c. read and process a user configuration based on the one or more profiles (page

4, paragraph 62);

d. select an identification of a particular GUI component (page 4, paragraph 67);

e. locate the identification in the mapping table (page 4, paragraph 67);

f. check a state of the particular GUI component (page 4, paragraph 69);

g. compare the state to the identified profile of the one or more profiles (page 4,

paragraph 69);

h. change the state if not in agreement with the identified profile of the one or

more profiles (page 4, paragraph 69);

Claim 25: <u>Oni</u> discloses a system of managing visibility of GUI components in an

application, comprising:

      a. initializing an application (page 3, paragraph 43);

      b. invoking a visibility manager (page 4, paragraph 62);

      c. displaying a user interface screen of the application according to an applied

one of the plurality of profiles (page 4, paragraph 65).

      <u>Oni</u> discloses invoking a visibility manager for a plurality of profiles and displaying

a user interface of the application wherein display of the GUI components are

determined by the visibility manager (page 4, paragraph 62), but does not explicitly

disclose providing a user interface of the visibility manager for selection of one of a

plurality of visibility states for each of at least a subset of the GUI components.

<u>Emmerichs</u> discloses a similar method of managing visibility of GUI components in an

application that further discloses selecting options in a security manager (visibility

manager) that control the visibility states of GUI objects accessed by a user through a

user interface. Therefore it would have been obvious to one having ordinary skill in the

art at the time the invention was made to provide a user interface of the visibility

manager in <u>Oni</u>, because providing a selection interface was recognized as part of the

ordinary capabilities of one skilled in the art. One would have been motivated to provide

a user interface of the visibility manager in order to provide access to GUI components

based on access privileges

      <u>Emmerichs</u> further discloses building the user interface screen with all of the

components set as visible, then invoking the visibility manager to determine, based on

the selected visibility states of the applied profile, which of the GUI components of the

built user interface screen are to be set as not visible and revise the built user interface

screen based on the determination and displaying the revised user interface screen of

the application (page 6, paragraph 54). Specifically Emmerichs discloses displaying all

of the user interface components without first determining whether the user has access

to each widget. After the user attempts to access the widget, the software checks the

availability of the widget and notifies a user with an error message if access to the

widget is denied. Therefore, it would have been obvious to one having ordinary skill in

the art to include these limitations in Oni. One would have been motivated to revise the

user interface based on the determination of the visibility state of the applied profile in

order to provide an initial GUI to the user while the profile is loading.

Claim 26: Oni and Emmerichs disclose a system of managing visibility of GUI

components in an application as in Claim 25 above, and Oni further discloses:

      a. reading the one or more profiles (page 4, paragraph 63);

      b. processing the one or more profiles (page 4, paragraph 63);

      c. reading and processing a user configuration based on the one or more profiles

(page 4, paragraph 62);

      d. activating a particular profile of the one or more profiles (page 4, paragraph

67);

Claim 28: Oni and Emmerichs disclose a system of managing visibility of GUI

components in an application as in Claim 26 above, and Oni further discloses:

    a. selecting an identification of a particular GUI component (page 4, paragraph

67);

    b. locating the identification in a mapping table (page 4, paragraph 67);

    c. checking a state of the particular GUI component (page 4, paragraph 69);

    d. comparing the state to the particular profile of the one or more profiles (page 4,

paragraph 69);

    e. changing the state if not in agreement with the particular profile of the one or

more profiles (page 4, paragraph 69);

Claim 29: Oni and Emmerichs disclose a system of managing visibility of GUI

components in an application as in Claim 26 above, and Oni further discloses:

    a. repeating locating the identification, checking the state, comparing the state,

and changing the state for any remaining identifications of additional GUI components

(page 4, paragraph 69).

Claim 30: Oni and Emmerichs disclose a system of managing visibility of GUI

components in an application as in Claim 26 above, and Oni further discloses:

    a. the state is visible or not visible (page 4, paragraph 69).

Claim 32: <u>Oni</u> and <u>Emmerichs</u> disclose a system of managing visibility of GUI

components in an application as in Claim 25 above, and <u>Oni</u> further discloses:

     a. reading the one or more profiles (page 4, paragraph 63);

     b. processing the one or more profiles (page 4, paragraph 63);

     c. reading and processing a user configuration based on the one or more profiles

(page 4, paragraph 62);

     d. selecting an identification of a particular GUI component (page 4, paragraph

67);

     e. locating the identification in a mapping table (page 4, paragraph 67);

     f. checking a state of the particular GUI component (page 4, paragraph 69);

     g. repeating locating the identification, checking the state, comparing the state,

and changing the state for any remaining identifications of additional GUI components

(page 4, paragraph 69).

Claim 34: <u>Oni</u> and <u>Emmerichs</u> disclose a system of managing visibility of GUI

components in an application as in Claim 33 above, and <u>Oni</u> further discloses:

     a. means for reading the one or more profiles upon initialization of the application

(page 4, paragraph 63);

     b. means for processing the one or more profiles (page 4, paragraph 63);

     c. means for reading and processing a user configuration based on the one or

more profiles (page 4, paragraph 62);

d. means for activating a particular profile of the one or more profiles (page 4, paragraph 67).

Claim 36: Oni discloses a system of managing visibility of GUI components, comprising:

a. starting up a visibility manager (page 4, paragraph 62);

b. an application building a data structure representing a user interface including a plurality of interface components (page 4, paragraph 62);

c. during the building of the data structure and responsive to the start-up of the visibility manager, the visibility manager selecting one of a plurality of interface profiles, each of the interface profiles indicating for each of a plurality of interface components a respective visibility instruction (page 4, paragraph 65);

Oni does not explicitly disclose subsequent to the building of the data structure, the application calling the visibility manager. Emmerichs discloses building a user interface with all of the components set as visible initially, then determining whether the user has access to a widget when the user attempts to access the widget. It is obvious that a visibility manager is called after the components built in Emmerichs, therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to include this feature in Oni. One would have been motivated to call the visibility manager subsequent to the building of the data structure in order to provide an initial GUI to the user while the profile is loading.

Emmerichs further discloses responsive to the call, the visibility manager

modifying the data structure in accordance with the visibility instructions of the selected

interface profile, the modified data structure representing a modified version of the user

interface which does not include all of the plurality of interface components and the

application displaying the modified version of the user interface. Specifically Emmerichs

discloses displaying all of the user interface components without first determining

whether the user has access to each widget. After the user attempts to access the

widget, the software checks the availability of the widget and notifies a user with an

error message if access to the widget is denied. One of the embodiments taught by

Emmerichs involves hiding widgets that the user does not have access to. Therefore, it

would have been obvious to one having ordinary skill in the art to include these

limitations in Oni. One would have been motivated to revise the user interface based on

the selected user profile in order to provide customization privileges to the user.

Claim 37: Oni and Emmerichs disclose a system of managing visibility of GUI

components in an application as in Claim 36 above, and Oni further discloses:

   a. the starting up of the visibility manager and the building of the data structure

by the application are performed in response to a start up of the application (page 5,

paragraph 70).

Claim 38: Oni and Emmerichs disclose a system of managing visibility of GUI

components in an application as in Claim 36 above, and Oni further discloses:

a. the selection of the selected interface profile is based on a user type

associated with a present use of the application (page 4, paragraph 62).

Claim 39: Oni and Emmerichs disclose a system of managing visibility of GUI

components in an application as in Claim 36 above, and Oni further discloses:

a. the selection of the selected interface profile is based on a user configuration

file (dynamic repository of user profiles) which identifies which of the plurality of

interface profiles a user associated with a present use of the application has set as

active (page 4, paragraph 67).

6. Claims 7, 15, 23, and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable

over Oni (US 2004/0133546) in view of Emmerichs (US 2003/0061482) and further in

view of Moshfeghi (US 6,476,833).

Claims 7, 15, 23, and 31: Oni discloses a method, system, data structure, and medium

embodying instructions as in Claims 5, 12, 21, and 28 above, further comprising the

mapping table comprises a plurality of identifications of GUI components (page 4,

paragraph 67). However, Oni does not explicitly disclose a corresponding plurality of

objects of an object-oriented and platform independent programming language.

Moshfeghi discloses a similar method, system, data structure, and medium embodying

instructions that further discloses the GUI components of the application are specified

with the Swing component set of the Java^{TM} Foundation Classes (Column 15, lines 19-

31). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made that the mapping table could comprise identifications of references to objects of an object-oriented and platform independent programming language. One would have been motivated to include Java software objects in view of the fact that Java is a widely used programming language throughout the Internet and World Wide Web (WWW).

Response to Arguments

8.   Applicant's arguments filed March 13, 2008 have been fully considered but they are not persuasive.

Claim 1: Applicant argues. "The Oni reference does not disclose or suggest an interface generation method that includes, for a particular display of a particular interface according to an applied profile, initially generating an interface with all GUI components set as visible and then revising the generated interface according to a profile." It is respectfully submitted that Oni modified by Emmerichs discloses the limitations of Claim 1. Emmerichs discloses displaying all of the user interface components without first determining whether the user has access to each widget. After the user attempts to access the widget, the software checks the availability of the widget and notifies a user with an error message if access to the widget is denied.

Conclusion

9.   Applicant's amendment necessitated the new ground(s) of rejection presented in

this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37

CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later

than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to OMAR ABDUL-ALI whose telephone number is

(571)270-1694. The examiner can normally be reached on Mon-Fri(Alternate Fridays

Off) 8:30 - 6:00 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Stephen Hong can be reached on 571-272-4124. The fax phone number for

the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information

system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

OAA   /Stephen S. Hong/
5/26/2008   Supervisory Patent Examiner, Art
                                                  Unit 2178